

Amendments to the Claims

This listing of claims will replace all prior versions, and listings of claims in the application:

1. *(previously presented)* A computer-implemented method for creating and/or modifying a dynamically update-able, searchable packet classification databank, comprising the steps of:

 receiving a collection of packet classification rules, each packet classification rule being represented as a plurality of bit positions;

 analyzing each of said plurality of bit positions to select a first bit position to partition said collection into at least two sets of siblings, wherein said analyzing includes applying at least one of empirical knowledge or a computed metric for each bit position to select said first bit position;

 selecting an index key corresponding to said first bit position;

 analyzing said plurality of bit positions to select a second bit position to partition said at least two sets of siblings into subsets of siblings; and

 selecting an index key corresponding to said second bit position.

2. *(previously presented)* The method of claim 1, further comprising the step of:

 selecting an index key corresponding to a third bit position to partition said subsets of siblings into further subsets.

3. *(previously presented)* The method of claim 1, further comprising the step of:

 repetitively partitioning said subsets of siblings into a hierarchy of subsets at a lower level until reaching a partition threshold.

4. (*previously presented*) The method of claim 3, wherein said partition threshold is predicated on a maximum number of rules residing in a subset of siblings at a respective lower level.

5. (*previously presented*) The method of claim 3, wherein said partition threshold is predicated on a maximum number of respective lower levels.

6. (*original*) The method of claim 1, wherein each sibling at a respective level has a substantially equivalent quantity of said packet classification rules.

7. (*previously presented*) The method of claim 1, wherein each of said analyzing steps include the steps of:

measuring a difference in cardinality at each bit position that has not been selected previously for a corresponding index key; and
computing an optimization parameter for each bit position.

8. (*previously presented*) The method of claim 7, wherein each of said analyzing steps further include the step of:

selecting an index key corresponding to a bit position having an optimization parameter closest to a predetermined value.

9. (*previously presented*) The method of claim 7, wherein each of said analyzing steps further include the step of:

selecting an index key corresponding to a first bit position determined to have an optimization parameter closest to a predetermined value in response to determining multiple bit positions having an optimization parameter closest to a predetermined value.

10. *(previously presented)* The method of claim 1, further comprising the step of:

receiving at least one packet classification rule within said collection that has one or more bit positions denoted as having a plurality of values.

11. *(previously presented)* The method of claim 10, wherein each of said analyzing steps further include the steps of:

measuring a difference in cardinality at each bit position that has not been selected previously for a corresponding index key; and

computing an optimization parameter for each bit position.

12. *(original)* The method of claim 11, wherein said computing an optimization parameter comprises:

determining an evenness of division for siblings at a respective level; and

determining an average cardinality.

13. *(previously presented)* The method of claim 1, further comprising the steps of:

receiving at least one packet classification rule within said collection that has two or more bit positions that denote a feature having a range of values; and

decomposing said at least one packet classification rule into two or more packet classification divisional rules, wherein said analyzing steps further include processing said divisional rules as part of said collection.

14. *(previously presented)* The method of claim 1, further comprising the step of:

manifesting a query key based on the selected index keys.

15. (*original*) The method of claim 14, further comprising the steps of:
 enabling addition and/or deletion of a packet classification rule in said
collection; and
 revising said query key in response to said addition and/or deletion of a
packet classification rule.
16. (*original*) The method of claim 15, further comprising the step of:
 performing said revising said query key on a periodically scheduled basis.
17. (*original*) The method of claim 15, further comprising the step of:
 performing said revising said query key on demand.
18. (*original*) The method of claim 14, further comprising the steps of:
 receiving a packet;
 applying said query key to said packet to produce a packet key; and
 searching said collection to detect a packet classification rule matching
said packet key.
19. (*previously presented*) The method of claim 18, further comprising the
steps of:
 detecting multiple packet classification rules matching said packet key;
and
 selecting a collision location key corresponding to a bit position to enable
partitioning of said multiple packet classification rules.
20. (*currently amended*) The method of claim 18, further comprising the steps
of:
 detecting multiple packet classification rules matching said packet key;
and

sequentially comparing each of said multiple packet classification rules with said packet to detect a matching rule.

21. (*original*) The method of claim 18, further comprising the step of:
enabling addition and/or deletion of a packet classification rule in said collection during said searching said collection.

22. (*previously presented*) A packet classification system, comprising:
a first memory for receiving a collection of packet classification rules, wherein each packet classification rule is represented as a plurality of bit positions; and
a mask constructor for selecting one or more index keys,
wherein each index key corresponds to a bit position that enables partitioning of said packet classification rules into at least two sets of siblings at a lower level,
wherein said mask constructor applies at least one of empirical knowledge or a computed metric for each bit position to select the bit position corresponding to each index key, and
wherein said mask constructor continues to select index keys to repetitively partition each set of siblings at a respective level into at least two sets of siblings at a lower level until reaching a partition threshold.

23. (*original*) The system of claim 22, wherein said mask constructor assembles said one or more index keys into a query key.

24. (*original*) The system of claim 23, further comprising:
a key extractor for applying said query key to produce a refined rule collection from said collection located within said first memory; and
a second memory for storing said refined rule collection.

25. (*original*) The system of claim 24, wherein said second memory is a content addressable memory.

26. (*original*) The system of claim 23, further comprising:
a key extractor for applying said query key to an incoming packet to produce a packet key.

27. (*original*) The system of claim 26, further comprising:
a packet classifier for applying said packet key to detect a packet classification rule matching said packet key.

28. (*original*) The system of claim 26, wherein said key extractor is a multiplexor, wherein said multiplexor is configured to select field descriptors from said packet based on said query key.

29. (*original*) The system of claim 28, wherein said multiplexor is a crossbar switch or a bit shifter.

30. (*previously presented*) A computer program product comprising a computer useable medium having computer readable program code means embedded in said medium for causing a computer to classify packet flows, comprising:

a first computer readable program code means for causing the computer to select one or more index keys,

wherein said first computer readable program code means selects each index key such that each index key corresponds to a bit position that enables partitioning of a set of packet classification rules into two or more sets of siblings at a lower level,

wherein said first computer readable program code means applies at least one of empirical knowledge or a computed metric for each location to select the bit position corresponding to each index key, and

wherein said first computer readable program code means continues to select index keys to repetitively partition each set of siblings at a respective level into at least two sets of siblings at a lower level until reaching a partition threshold; and

a second computer readable program code means for causing the computer to assemble said one or more index keys into a query key.

31. *(previously presented)* A computer-implemented method for creating and/or modifying a dynamically update-able, searchable packet classification databank, comprising the steps of:

receiving a collection of packet classification rules, each packet classification rule being represented as a plurality of location coordinates;

selecting an index key based on a common location coordinate among said packet classification rules at a first level, such as to enable partitioning of said collection into two or more siblings at a second level, wherein the coordinate value of said common location coordinate represents a feature whereby the composition of each sibling contains packet classification rules possessing a common feature; and

selecting an index key based on a second common location coordinate among said packet classification rules at said second level, such as to enable partitioning of at least one of said two or more siblings at said second level into two or more siblings at a third level,

wherein each of said selecting an index key step comprises the steps of:

measuring a difference in cardinality at each location coordinate that has not been selected previously as an index key; and

computing an optimization parameter for each location coordinate.

32. *(previously presented)* The method of claim 31, wherein each of said selecting an index key step further comprises the step of:

selecting an index key corresponding to a location coordinate having an optimization parameter closest to a predetermined value.

33. *(previously presented)* The method of claim 31, wherein each of said selecting an index key step further comprises the step of:

selecting an index key corresponding to a first location coordinate determined to have an optimization parameter closest to a predetermined value in response to determining multiple location coordinates having an optimization parameter closest to a predetermined value.

34. *(previously presented)* A computer-implemented method for creating and/or modifying a dynamically update-able, searchable packet classification databank, comprising the steps of:

receiving a collection of packet classification rules, each packet classification rule being represented as a plurality of location coordinates;

receiving at least one packet classification rule within said collection that has one or more location coordinates denoted as having a plurality of values;

selecting an index key based on a common location coordinate among said packet classification rules at a first level, such as to enable partitioning of said collection into two or more siblings at a second level, wherein the coordinate value of said common location coordinate represents a feature whereby the composition of each sibling contains packet classification rules possessing a common feature; and

selecting an index key based on a second common location among said packet classification rules at said second level, such as to enable partitioning of at least one of said two or more siblings at said second level into two or more siblings at a third level,

wherein each of said selecting an index key step comprises the steps of:

measuring a difference in cardinality at each location coordinate that has not been selected previously as an index key; and
computing an optimization parameter for each location coordinate.

35. *(previously presented)* The method of claim 34, wherein said computing an optimization parameter comprises:
determining an evenness of division for siblings at a respective level; and
determining an average cardinality.